(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 November 2001 (22.11.2001)

PCT

(10) International Publication Number
WO 01/88817 A1

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US01/15672

(22) International Filing Date: 15 May 2001 (15.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/204,686	17 May 2000 (17.05.2000)	US
09/805,551	13 March 2001 (13.03.2001)	US
09/805,552	13 March 2001 (13.03.2001)	US

(71) Applicant *(for all designated States except US)*: **CENTENNIAL TECHNOLOGIES, INC.** [US/US]; 7 Lopez Road, Wilmington, MA 01887 (US).

(72) Inventors; and

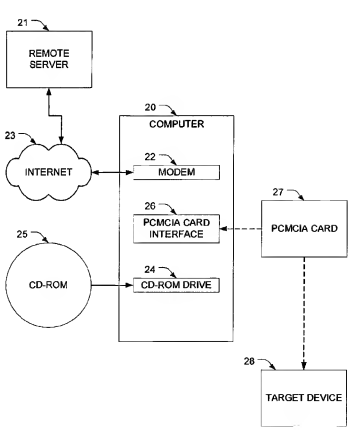
(75) Inventors/Applicants *(for US only)*: **LAMBERT, Grady** [US/US]; 26 New Pond Road, Groton, MA 01450 (US). **MCCORMICK, Thomas** [US/US]; 9 Noble Drive, Chelmsford, MA 01824 (US).

(74) Agents: **CATAN, Mark, A.** et al.; Lyon & Lyon LLP, 633 West Fifth Street, Suite 4700, Los Angeles, CA 90071-2066 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: FEE MANAGEMENT FOR PROGRAMMING OF ELECTRONIC DEVICES



(57) Abstract: A program and/or data is written into a memory in a plug-in module (27) or into a standalone device only when a payment is made. In some embodiments, the programs and/or data is stored in an encrypted form on CD-ROM (25) or other types of media, and a decryption code is obtained via the Internet (23). In other embodiments, the data is obtained via the Internet. Either decryption of the data or writing of the data is inhibited until appropriate payment is made, and access to the write data is prevented after the write operation has occurred.

WO 01/88817 A1

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

S P E C I F I C A T I O N

TITLE

FEE MANAGEMENT FOR PROGRAMMING OF ELECTRONIC DEVICES

BACKGROUND OF THE INVENTION

1. Field of the Invention

The field of the present invention relates to the programming of electronic devices. More specifically, the field of the present invention relates to upgrading programs that are executed by electronic devices and to loading data files into electronic devices.

2. Background

Many electronic devices are designed so that their program and data files can be upgraded or reconfigured after they have been released in the field. In some cases, vendors provide these programs and data files for free, and make the files accessible for downloading from a remote host via the Internet. For example, the Microsoft™ Internet Explorer program can be downloaded onto a user's computer for free via the Internet. Once the program has been downloaded, it can be executed on the computer onto which it was downloaded, or copied onto other computers and executed on those other computers.

In other cases, the vendors charge a fee for their programs and data. Typically, the fee is charged before the program or data is transferred from the remote host to the local computer. Here again, once the program has been downloaded to the local computer, it can be executed on the local computer, copied onto other computers, and executed on those other computers as well. As a result, even though the

vendor has collected only one fee for their program or data, the program or data may be used by multiple end-users on multiple computers.

When the data or program is large and/or the bandwidth of the local computer's Internet connection is limited, downloading programs and data over the Internet can be annoyingly slow. For example, downloading a 100 Mbyte file over a 56 Kbps modem link takes over 5 hours, during which time a phone line is usually tied up. One way to deal with this shortcoming is to distribute the data on CD-ROMs, which can store over 500 Mbytes of data. This arrangement works well for programs and data that are distributed for free (such as the program for accessing the America Online™ service), but usually not for vendors who wish to get paid for each installation of a program.

U.S. Patent No. 5,809,145, which is incorporated herein by reference, describes a system that enables a vendor to distribute data on CD-ROMs, yet still exact payment for use of the data. Note that the term "data", as used herein, includes program files and/or information that is used by programs. The '145 patent teaches encrypting the data on the CD-ROM, and selling a decryption key to the user. The purchase of the decryption key may be transacted over the Internet. Once a decryption key is purchased, the computer decrypts the data and copies the decrypted version onto the computer's hard disk. A shortcoming of this system, however, is that once the data is stored on the hard drive in its decrypted form, it can subsequently be copied onto another computer. As a result, a vendor may not receive payment for each copy that is eventually made.

In some situations, a computer is used to read data from a data source (e.g., from the Internet or from a CD-ROM), but the computer is not the ultimate destination for the data that has been read. Instead, the destination for the data is an auxiliary device that is distinct from the computer, and usually capable of operating in a standalone mode (i.e., independent of the computer). With these systems, the computer is merely used as a means to get the data into the auxiliary device.

With a first class of such auxiliary devices, the computer sends the data directly into the auxiliary device via a wired or wireless communication link. Examples of this class of auxiliary devices include personal digital assistants (PDAs) and MP3 players. For example, application programs for PDAs (e.g., a Palm™ PDA) can be downloaded from a remote host to a local computer via the Internet, and subsequently transferred from the local computer into a PDA that is connected to the local computer via a cable and/or cradle. In this case, the ultimate destination for the program would be the PDA – not the local computer. Similarly, MP3 data files can also be downloaded from a remote host to a local computer via the Internet, and subsequently transferred to an MP3 audio player (e.g., a Rio™ MP3 player) that is connected via a cable to the local computer. In this case, the ultimate destination for the data file would be the MP3 player.

For this first class of auxiliary devices, it is known to place data on the Internet, and make that data available for downloading in exchange for a fee. The data is downloaded into a local computer, and subsequently transferred into the auxiliary device. The user is charged

for the data when it is downloaded to the user's local computer (e.g., by charging a credit card account). But controlling access to the data at the point of downloading to the local computer from the Internet is not ideal, because the data could subsequently be transferred from the local computer into more than one target device (e.g., into multiple MP3 players, or multiple PDAs. As a result, a user could obtain multiple copies of a program or an MP3 file even though he has only paid for one copy. Controlling access to encrypted data on a CD-ROM using a decryption process that writes an unencrypted version onto the hard disk of a local computer is also not ideal for the same reasons.

With a second class of auxiliary devices, a computer is used to program a plug-in module (e.g., a PCMCIA card), and the plug-in module is then unplugged from the computer and plugged into the auxiliary device. The auxiliary device then uses the data that has been written into the plug-in module. In these systems, the ultimate destination for the data is the plug-in module.

Traditionally, upgrades to plug-in upgradable hardware have been implemented by distributing new plug-in modules to the end user, and asking the end-user to replace the old plug-in module with the new one and to return the old plug-in module to the manufacturer or an authorized service center. This approach can be used to insure that the vendor is paid for each copy of the upgrade. However, due to the cost of the plug-in modules themselves, it can be expensive when an end user fails to return the old plug-in module for reprogramming. In addition, many people do not bother to

order upgrades, which can result in the use of outdated data by users, and lost revenue potential for manufacturers.

The inventors have recognized a need to upgrade programs and/or data in standalone devices and in plug-in upgradable devices, while ensuring that vendors are paid for each system that is upgraded.

SUMMARY OF THE INVENTION

The present invention relates to writing data into memories in plug-in modules or in other target devices.

One aspect of the present invention is directed to a method of writing data into a memory contained in a plug-in module. In this method, a plug-in module is accepted in a connector configured to interface with the plug-in module. First data is read from a data storage device, and an access control code that permits generation of second data based on the first data is obtained. Second data is then generated based on the first data and the obtained access control code. The memory contained in the plug-in module is written to based on the generated second data, and access to the second data is subsequently prevented. A notification is provided after completion of the writing.

Another aspect of the present invention is directed to a method of writing data into a memory contained in a target device. In this method, a communication link is established with the target device. First data is read from a data storage device, and an access control code that permits generation of second data based on the first data is obtained. Second data is then generated based on the first data and the obtained access control code. Signals are sent, via the communication link, into the target device.

These signals include instructions for writing to the target device and are based on the generated second data. Access to the second data is prevented after execution of the sending step, and notification is provided after completion of the sending step.

Another aspect of the present invention is directed to a method of writing data into a memory contained in a plug-in module. In this method, the plug-in module is accepted in a connector configured to interface with the plug-in module. Payment information is provided to a remote server, and authorization to proceed is received from the remote server. Data is received from the remote server. Based on the received data, data is written into the memory contained in the plug-in module. Further access to the received data is prevented after the data has been written. Notification is provided after completion of the writing step.

Another aspect of the present invention is directed to a method of writing data into a memory contained in a target device. In this method, a communication link is established with the target device, and payment information is provided to a remote server. An authorization to proceed is received from the remote server, and data is received from the remote server. Signals are sent, via the communication link, into the target device. The signals are based on the data received in the receiving step, and they include instructions for writing to the target device. Access to the data received in the receiving step is prevented after execution of the sending step. Notification is provided after completion of the sending step.

Another aspect of the present invention is directed to a method of using a computer to write data into a memory for

an electronic device. In this method, data is loaded into the computer from a data source, and a removable connection is established between the computer and the memory for the electronic device. A determination is made to see if payment has been made. In response to that determination, data is written into the memory for the electronic device, via the connection. The removable connection between the computer and the memory for the electronic device is then removed. The electronic device is then operated independently of the computer using the written data.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic hardware block diagram of a first embodiment and a second embodiment of the present invention.

FIG. 2 is a functional block diagram for the first embodiment.

FIG. 3 is a flowchart of the processes implemented in the first embodiment.

FIG. 4 is a functional block diagram for the second embodiment.

FIG. 5 is a flowchart of the processes implemented in the second embodiment.

FIG. 6 is a schematic hardware block diagram of a third embodiment and a fourth embodiment of the present invention.

FIG. 7 is a functional block diagram for the third embodiment.

FIG. 8 is a functional block diagram for the fourth embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the first and second embodiments illustrated in FIG. 1, a computer 20 includes a modem 22, which enables the computer 20 to establish a communication link with a remote server 21 via the Internet 23. The computer 20 also includes a CD-ROM drive 24 that enables the computer to read data from a CD-ROM 25, and a PCMCIA card interface 26 that enables the computer to write data onto a PCMCIA card 27 that has been inserted into a PCMCIA slot in the computer 20. After data has been written onto the PCMCIA card 27, the PCMCIA card 27 may be removed from the computer 20 and inserted into the target device 28. The hardware for the illustrated computer 20 is widely available off-the-shelf from a variety of computer manufacturers including HP, IBM, Compaq, and Apple.

In alternative embodiments, other types of plug-in cards or modules may be substituted for the PCMCIA card 27, as long as the interface 26 is modified to match the card or module. These alternative plug-in cards or modules may be custom-designed to interface with a particular piece of hardware, or may be designed to comply with an industry-standard interface. Similarly, other types of data storage media may also be substituted for the CD-ROM 25. Examples of suitable substitute media include floppy disks, hard disk drives, and magnetic tape. The storage media may also be located at a remote location, and linked to the computer via a suitable network (e.g. the Internet). Of course, when an alternative media is used, the CD-ROM drive 24 must be replaced with the appropriate hardware for interfacing with that media.

In alternative embodiments, the Internet communication link between the computer 20 and the remote server 21 may be replaced with an alternative communication link. For example, a direct dial-up modem link may be established between the computer 20 and the remote server 21 via the modem 22, or a hardwired connection (using, e.g., wire or fiber optic cable) may be established between the computer 20 and the remote server 21. Of course, when substitutions for any of the components illustrated in FIG. 1 are made, appropriate modifications to the description that follows will be required, as will be appreciated by persons skilled in the relevant arts.

The purpose of the first embodiment described in connection with FIGS. 1-3 is to enable authorized users to load a program (or data to be used by a program) into a PCMCIA card 27 based on data contained in the CD-ROM 25, and to prevent unauthorized copies of the program (or data) from being made. One preferred approach to accomplishing this is to encrypt the data on the CD-ROM 25, and to prevent the decryption of the data and/or the writing of the data to the PCMCIA card 27 until the user has paid for the data. In this approach, it is envisioned that the data provider will distribute the CD-ROMs for free to a relatively large audience, and encourage them to buy the information that they wish to use.

FIG. 2 is a schematic block diagram illustration of some of the processes implemented in the computer 20, and the flow of data through the system that occurs when those processes are implemented. It is best understood when viewed in connection with FIG. 3, which is a flowchart of the implemented processes. In the following discussion,

ordinary reference numbers can be found in FIG. 2, and reference numbers preceded with an "S" can be found in FIG. 3.

Preferably, the illustrated processes are implemented by an application program that is executed by the computer 20. This program may be started in any conventional manner, such as by double-clicking on an appropriate icon on a display, or by inserting a CD-ROM into the CD-ROM drive 24 on the computer 20 and closing the drive door. The application program preferably includes a plurality of software modules including a user interface 31, a decryption process 32, and a card programming subroutine 33.

In some embodiments, both the application program this is executed on the local computer 20 and the data that will be written into the PCMCIA card 27 are stored on the same CD-ROM 25, but in distinct files. In these embodiments, the program instructs the computer 20 to read the data file from the CD-ROM 25 and to generate corresponding write data for the PCMCIA card 27. Optionally, two or more data files may be included on the CD-ROM 25, in which case the program would select a desired one of the data files to generate write data for the PCMCIA card 27 (based, e.g., on a user's selection). Because the program and data both reside on the same CD-ROM, there is no need to swap discs in the middle of the program. In these embodiments, the data files are preferably encrypted to prevent unauthorized access.

In alternative embodiments, the data that is to be written into the PCMCIA card 27 is embedded within the same file that is being executed by the computer. In this case, only a single file is needed on the CD-ROM 25 to generate write data for the PCMCIA card 27. Optionally, more than

one such combined program/data file may be included on the CD-ROM 25, and the user can select which program to run. This arrangement also eliminates the need to swap discs in the middle of the program. In these embodiments, the portions of the combined program/data files that control the data extraction and writing process are preferably encrypted to prevent unauthorized access.

In another alternative embodiment, the application program is stored on the computer's hard drive (not shown), and the data that is to be written into the PCMCIA card 27 is stored on the CD-ROM 25. Optionally, more than one data file may be included on the CD-ROM 25, and the user can select which file to use. In this embodiment, the data files are preferably encrypted to prevent unauthorized access. In a less preferred embodiment, the program and data may be stored on separate CD-ROMS, but disc swapping would be required.

Execution of the program begins at step S51, where the system checks that the CD-ROM with the write data and the PCMCIA card 27 are both inserted into their respective bays on the computer. If either the CD-ROM 25 or the PCMCIA card 27 is not inserted, an appropriate prompt is displayed to the user. In embodiments where the program and data are incorporated into the same CD-ROM 25, the step of checking for the CD-ROM may be omitted.

When the CD-ROM 25 contains more than one file, the user is prompted to select a desired file for writing to the PCMCIA card 27 in step S52, and the user's selection is noted by the system. When the CD-ROM 25 contains only one file, this step is preferably skipped.

In step S53, payment for the selected file is processed. Preferably, this includes the step of determining a price for transferring the file from the CD-ROM 25 to the PCMCIA card 27. The price may be determined by referencing price information stored on the CD-ROM 25 itself, preferably in a non-encrypted format. Alternative ways to determine the price include obtaining a price for the desired file from the Internet by sending an appropriate request to the remote server 21 via the Internet 23, and waiting for a response from the remote server. Next, the price for the selected file is presented to the user via the user interface 31, and the user is given an opportunity to pay for the selected file. Any desired payment plan may be implemented, such as pay-per-download, pay per Mbyte, or a flat annual or monthly rate for unlimited use.

Preferably, the payments are processed (or prior payments are verified) by communicating with the remote server 21 via the Internet 23 in any conventional manner, using an appropriate user interface (e.g., by charging a credit card provided by the user, or by charging an account that has been pre-established with the user). Optionally, the server may confirm the identity of the local computer's user (using, for example, a password or a cookie) and reject unauthorized users. After the payment information has been received by the remote server 21, the remote server 21 sends a decryption key to the local computer 20 via the Internet 23. In alternative embodiments, the decryption key may be provided in other ways such as via fax, telephone, email, etc. In step S54, this decryption key is accepted and provided to the decryption process 32.

Preferably, the decryption process is configured to treat the encrypted file as a plurality of blocks, and to decrypt one block at a time. The size of each of these blocks may be selected so that when the encrypted block is eventually decrypted, the resulting decrypted data will be an integer multiple of a block size associated with the hardware in the destination PCMCIA card 27. For example, if the block size for the destination PCMCIA card 27 is 1024 bytes, the size of the decrypted blocks of data would preferably be 1024, 2048, or 3072 bytes, etc.

In step S55, the decryption process 32 uses the decryption key received in step S54 to decrypt the first block of data from the selected file in the CD-ROM 25, resulting in a corresponding block of decrypted data.

In some embodiments, step S56 is implemented in order to provide an additional degree of assurance that programming of the PCMCIA card 27 has been authorized. In these embodiments, the application program sends an inquiry to the remote server 21 via the Internet 23 to determine if programming of the PCMCIA card 27 has been authorized. Preferably, the remote server 21 is programmed to provide authorization only when payment for programming has been made. If authorization to proceed is received from the remote server 21, processing continues to step S57. But if authorization to proceed is not received from the remote server 21, programming of the PCMCIA card 27 will be aborted. Note that while step S56 is illustrated as occurring between the read and write operations in the loop, the authorization check could be moved to another position in the loop (e.g., after the write operation).

In alternative embodiments, more complicated authorization verification schemes may be implemented. For example, each block of the data file may be encrypted individually, so that each block requires its own decryption key. In this case, once the decryption of a single block has been completed, the system requests the decryption key for the next block from the remote server 21. If the next decryption key is received, the decryption process can continue. But if the next decryption key is not received, the decryption process terminates before all the desired data is written onto the PCMCIA card 27.

In other alternative embodiments, step S56 may be omitted altogether, and the program may rely on the receipt of a decryption key (in step S54) as an indication of authorization to program the PCMCIA card 27.

In step S57, the current block of decrypted data is written into the PCMCIA card 27. Preferably, this is accomplished by invoking a suitable card programming subroutine 33, which sends appropriate instructions to the hardware interface 35. The hardware interface 35 then sends appropriate signals into the PCMCIA card 27 to program the card. The card programming subroutine 33 and the hardware interface 35 may be implemented using any of a variety of well known techniques for programming a block of data in a PCMCIA card. The particulars of these techniques may depend on the specific model of PCMCIA card being programmed.

After the current block of data has been written into the PCMCIA card 27, processing proceeds to step S58, where a test is performed to determine whether any additional data from the selected file needs to be written into the PCMCIA card 27. If there is additional data, control of the

program returns to step S55 for processing of the next block of data. If there is no additional data, control of the program passes to step S59.

When the data from the selected file is divided into a plurality of blocks, the arrangement of steps S55-S58 into a loop results in an interleaving of the decryption and writing processes. For example, if the data is divided into 3 blocks, the temporal sequence of events would be as follows: (1) decrypt the first block; (2) write the first block; (3) decrypt the second block; (4) write the second block; (5) decrypt the third block; and (6) write the third block.

Preferably, access to each block of data is prevented after that block is written into the PCMCIA card 27. This may be accomplished, for example, by erasing each block of data after execution of the write operation, or by overwriting the previously decoded block of data with new data when the next block of data is decoded. When this approach is used, there will never be a time when the entire file exists in the memory of the computer 20 in a decrypted form. And when a complete decrypted version of the file does not exist, it becomes much more difficult to make an unauthorized copy of the decrypted data. In a less preferred embodiment, the decryption process treats the encrypted file as a single large block. But using a single large block is not as advantageous as using multiple blocks, because a complete decrypted version of the file would exist (at least for a short while).

In step S59, a notification that the write process has been completed is provided to the user (e.g., by illuminating an indicator, by displaying an appropriate text

message, by generating a beep or another sound, or in any other appropriate manner). Preferably, the user is prompted to remove the PCMCIA card 27 and to insert it into the target device. Depending on the particular PCMCIA card 27 being programmed, the user may be instructed to shut down the computer 20 before removing the PCMCIA card 27. Once the PCMCIA card 27 has been installed in the target device, the program and/or data that was just written to the card can be accessed by the target device.

Optionally, the files on the CD-ROM 25 may be stored in a compressed format. When this option is implemented, the data from the CD-ROM 25 must be decompressed as well as decrypted.

The purpose of the second embodiment described in connection with FIGS. 4 and 5 is to enable authorized users to load a program (or data to be used by a program) into a PCMCIA card 27 based on data obtained from a remote server, preferably via the Internet 23, and to prevent unauthorized copies from being made. Writing of the data to the PCMCIA card 27 is not permitted unless the user has paid for the data. This embodiment is preferably used when the volume of data to be written into the PCMCIA card 27 is small enough so that the transmission delay will not be too annoying.

FIG. 4 is a schematic block diagram illustration of some of the processes implemented in the computer 20, and the flow of data through the system that occurs when those processes are implemented. It is best understood when viewed in connection with FIG. 5, which is a flowchart of the implemented processes. In the following discussion, ordinary reference numbers can be found in FIG. 4, and

reference numbers preceded with an "S" can be found in FIG. 5.

Operation of the elements 20-33 in this embodiment are similar to the corresponding elements in the first embodiment, except that the data for programming the PCMCIA card 27 is obtained from the remote server 21 instead of from the CD-ROM 25, and decryption is not required.

As in the first embodiment, the application program may be stored in any conventional manner (e.g., on a CD-ROM or on a hard disk) and launched in any conventional manner. Execution of the program begins at step S151, where the system checks that the PCMCIA card 27 is inserted into its bay on the computer. If the PCMCIA card 27 is not inserted, an appropriate prompt is displayed to the user.

When more than one file is available for downloading from the remote server 21, the user is prompted to select a desired file for transferring to the PCMCIA card 27 in step S152, and the user's selection is noted by the system. When only one file is available for downloading, this step is preferably skipped.

In step S153, payment for the selected file is processed. Preferably, this includes the step of determining a price for transferring the file to the PCMCIA card 27 by sending an appropriate request to the remote server 21 via the Internet 23, and waiting for a response from the remote server. Next, the price for the selected file is presented to the user via the user interface 31, and the user is given an opportunity to pay for the selected file. If the user agrees to the price, the user authorizes payment (e.g., by providing a credit card number or clicking

on an icon that authorizes payment to a debit or credit account), and the computer sends information to the remote server 21 to indicate that payment has been authorized.

In some embodiments, the remote server accepts the payment information and sends a verification that payment has been received back to the computer 20. This verification is interpreted by the computer 20 as an authorization to proceed. In other embodiments, the remote server accepts the payment information and immediately begins sending the requested data back to the computer 20. In these embodiments, receipt of the requested data is interpreted by the computer 20 as an authorization to proceed.

After payment has been made, the remote server 21 begins to send the selected file to the computer 20 via the Internet 23. Preferably, the remote server 21 is configured to treat the file as a plurality of blocks, to send one block to the computer 20 at a time, and to pause after each block has been sent. The size of each of these blocks may be selected in a manner similar to the first embodiment described above.

In step S155, the block of data arriving from the server is accepted and stored in a temporary buffer.

Optionally, step S156 may be implemented to provide an additional degree of assurance that programming of the PCMCIA card 27 has been authorized. Its operation is similar to step S56, which is described above in connection with the first embodiment. Optionally, authorization to continue may be implemented implicitly by having the server restart data transmission after a pause in transmission.

Step S156 may also be omitted altogether, and the program may rely on the receipt of data (in step S155) as an indication of authorization to program the PCMCIA card 27.

In step S157, the current block of data is written from the temporary buffer into the PCMCIA card 27. Preferably, this is accomplished by invoking a suitable card programming subroutine 33, which sends appropriate instructions to the hardware interface 35, as described above in connection with the first embodiment.

After the current block of data has been written into the PCMCIA card 27, processing proceeds to step S158, where a test is performed to determine whether any additional data from the selected file needs to be written into the PCMCIA card 27. If there is additional data, control of the program returns to step S155 for processing of the next block of data. If there is no additional data, control of the program passes to step S159, where a notification is provided to the user that the write process has ended (similar to step S159 described above in connection with the first embodiment). The newly programmed PCMCIA card 27 may then be installed in the target device.

When the data from the selected file is divided into a plurality of blocks, the operation of steps S155-S158 results in an interleaving of the reading and writing processes. For example, if the data is divided into 3 blocks, the temporal sequence of events would be as follows: (1) read the first block; (2) write the first block; (3) read the second block; (4) write the second block; (5) read the third block; and (6) write the third block.

As in the first embodiment, access to each block of data is preferably prevented after that block is written into the PCMCIA card 27. This may be accomplished, for example, by erasing the temporary buffer after execution of the write operation, or by overwriting the previously decoded block of data in the temporary buffer with new data whenever a new block of data is decoded. When this approach is used, there will never be a time when the entire file exists in the memory of the computer 20, which makes it more difficult to make an unauthorized copy of the data.

FIG. 6 is a schematic hardware block diagram of the third and fourth embodiments, and FIGS. 7 and 8 are functional block diagram for the third and fourth embodiments respectively. The third and fourth embodiments are similar to the first and second embodiments described above, except that the PCMCIA card and interface are replaced with a target device 128 and a data port 126. The target device 128 includes a memory 127, which may be permanently installed within the target device 128, or may be removably connected in the target device 128 using a suitable connector. As in the first and second embodiments, each of the components 20-25 may be replaced with various substitutes.

Preferably, one of the standard ports of the computer 20 (e.g., an RS-232 serial port, a Centronics parallel port, or a USB port) serves as the data port 126. An interface cable connects the data port 126 with the target device 128. The configuration of this interface between the computer 20 and the target device 128 will depend on the particular target device being used. For example, if a Palm™ PDA is being used, the interface will have a cradle that mates with

the PDA at one end, and a 9 pin serial connector at the other end. The computer and the target device 128 establish a communication link with each other via this interface. Alternatively, a wireless communication link may be used (using, e.g., infrared or radio-frequency wireless signals).

In the first and second embodiments, a plug-in module is reprogrammed while it is connected to a computer, and the plug-in module is subsequently removed from the computer and plugged into the target device. The end result is to modify either the program memory or the data memory of the target device.

In contrast, the target device in the third and fourth embodiments modifies its own program memory or data memory, in accordance with write instructions that are received via the communication interface. Operation of these embodiments is similar to operation of the first and second embodiments, except that instead of writing data to a PCMCIA card 27 (shown in FIG. 2) by sending appropriate signals to a PCMCIA card interface 35 (shown in FIG. 2), the third and fourth embodiments write data to a memory 127 in the target device 128 by sending appropriate signals to a device interface 135. Details regarding the signal formats, the instructions that must be sent to the target device 128 to initiate the write operation, and the device interface 135 will depend on the particular target device 128 being used.

The processes implemented in the third embodiment are similar to the processes described in connection with FIG. 3, except that (1) instead of prompting a user to insert or remove a PCMCIA card, the user is prompted to attach or detach the target device 128; and (2) instead of writing data directly to a PCMCIA card, data is written to the

memory 127 in the target device 128 by sending appropriate write instructions to the target device via device interface 135. Similarly, the processes implemented in the fourth embodiment are similar to the processes described in connection with FIG. 5, with the same two exceptions.

While the present invention has been explained in the context of the preferred embodiments described above, it is to be understood that various changes may be made to those embodiments, and various equivalents may be substituted, without departing from the spirit or scope of the invention, as will be apparent to persons skilled in the relevant art.

WHAT IS CLAIMED IS:

1. A method of writing data into a memory contained in a plug-in module, the method comprising the steps of:

accepting the plug-in module in a connector configured to interface with the plug-in module;

reading first data from a data storage device;

obtaining an access control code that permits generation of second data based on the first data;

generating second data based on the first data read in the reading step and the access control code obtained in the obtaining step;

writing to the memory contained in the plug-in module based on the second data generated in the generating step;

preventing access to the second data after execution of the writing step; and

providing a notification after completion of the writing step.

2. The method of claim 1, wherein the first data comprises encrypted data, the access code comprises a key for decrypting the encrypted data, and the second data comprises a decrypted version of the first data, and wherein the generating step comprises the step of decrypting the first data, using the key, so that the second data is formed.

3. The method of claim 2, further comprising the step of inserting the plug-in module into a target device.

4. The method of claim 2, wherein the generating step comprises the steps of periodically obtaining

authorization to continue via the Internet, and aborting the generating step if authorization to continue is not obtained when expected.

5. The method of claim 2, wherein the plug-in module comprises a PCMCIA card.

6. The method of claim 2, wherein the reading, generating, writing, and preventing steps for a first block of data are all performed prior to the reading, generating, writing, and preventing steps for a second block of data.

7. The method of claim 2, wherein the preventing step comprises the step of erasing the second data or overwriting the second data.

8. The method of claim 2, wherein the data storage device comprises a CD-ROM.

9. The method of claim 2, wherein, in the obtaining step, the access control code is obtained via the Internet.

10. The method of claim 2, wherein the access control code obtained in the obtaining step is provided by a remote server in response to receipt of a payment.

11. The method of claim 2, wherein the reading, generating, writing, and preventing steps for a first block of data are all performed prior to the reading, generating, writing, and preventing steps for a second block of data,

wherein the preventing step comprises the step of erasing the second data or overwriting the second data,

wherein the data storage device comprises a CD-ROM,

wherein, in the obtaining step, the access control code is obtained via the Internet, and

wherein the access control code obtained in the obtaining step is provided by a remote server in response to receipt of a payment.

12. A method of writing data into a memory contained in a target device, the method comprising the steps of:

establishing a communication link with the target device;

reading first data from a data storage device;

obtaining an access control code that permits generation of second data based on the first data;

generating second data based on the first data read in the reading step and the access control code obtained in the obtaining step;

sending signals, via the communication link, into the target device, wherein the signals comprise instructions for writing to the target device and are based on the second data generated in the generating step;

preventing access to the second data after execution of the sending step; and

providing a notification after completion of the sending step.

13. The method of claim 12, wherein the first data comprises encrypted data, the access code comprises a key for decrypting the encrypted data, and the second data comprises a decrypted version of the first data, and wherein the generating step comprises the step of decrypting the

first data, using the key, so that the second data is formed.

14. The method of claim 13, further comprising the step of disconnecting the communization link with the target device so that the target device can be operated independently in a standalone mode.

15. The method of claim 13, wherein the generating step comprises the steps of periodically obtaining authorization to continue via the Internet, and aborting the generating step if authorization to continue is not obtained when expected.

16. The method of claim 13, wherein the reading, generating, sending, and preventing steps for a first block of data are all performed prior to the reading, generating, sending, and preventing steps for a second block of data.

17. The method of claim 13, wherein the preventing step comprises the step of erasing the second data or overwriting the second data.

18. The method of claim 13, wherein the data storage device comprises a CD-ROM.

19. The method of claim 13, wherein, in the obtaining step, the access control code is obtained via the Internet.

20. The method of claim 13, wherein the access control code obtained in the obtaining step is provided by a remote server in response to receipt of a payment.

21. The method of claim 13, wherein the reading, generating, sending, and preventing steps for a first block of data are all performed prior to the reading, generating, sending, and preventing steps for a second block of data,

wherein the preventing step comprises the step of erasing the second data or overwriting the second data,

wherein the data storage device comprises a CD-ROM,

wherein, in the obtaining step, the access control code is obtained via the Internet, and

wherein the access control code obtained in the obtaining step is provided by a remote server in response to receipt of a payment.

22. A method of writing data into a memory contained in a plug-in module, the method comprising the steps of:

accepting the plug-in module in a connector configured to interface with the plug-in module;

providing payment information to a remote server,

waiting to receive an authorization to proceed from the remote server in response to the payment information provided in the providing step;

receiving data from the remote server;

writing data into the memory contained in the plug-in module based the data received in the receiving step;

preventing access to the data received in the receiving step after execution of the writing step; and

providing a notification after completion of the writing step.

23. The method of claim 22, further comprising the step of inserting the plug-in module into a target device.

24. The method of claim 22, wherein the plug-in module comprises a PCMCIA card.

25. The method of claim 22, wherein the receiving, writing, and preventing steps for a first block of data are all performed prior to the receiving, writing, and preventing steps for a second block of data.

26. The method of claim 22, wherein the data received in the receiving step comprises the authorization in the waiting step.

27. The method of claim 22, wherein the notification providing step comprises the step of notifying a user that the plug-in module may be removed from the connector.

28. The method of claim 22, wherein, in the providing step, the payment information is provided to the remote server via the Internet, and

wherein, in the receiving step, the data is received from the remote server via the Internet.

29. The method of claim 22, wherein the preventing step comprises the step of erasing the data received in the receiving step or overwriting the data received in the receiving step.

30. The method of claim 22, wherein the receiving, writing, and preventing steps for a first block of data are all performed prior to the receiving, writing, and preventing steps for a second block of data,

wherein the notification providing step comprises the step of notifying a user that the plug-in module may be removed from the connector,

wherein, in the providing step, the payment information is provided to the remote server via the Internet,

wherein, in the receiving step, the data is received from the remote server via the Internet, and

wherein the preventing step comprises the step of erasing the data received in the receiving step or overwriting the data received in the receiving step.

31. A method of writing data into a memory contained in a target device, the method comprising the steps of:

establishing a communication link with the target device;

providing payment information to a remote server,

waiting to receive an authorization to proceed from the remote server in response to the payment information provided in the providing step;

receiving data from the remote server;

sending signals, via the communication link, into the target device, wherein the signals comprise instructions for writing to the target device and are based on the data received in the receiving step;

preventing access to the data received in the receiving step after execution of the sending step; and

providing a notification after completion of the sending step.

32. The method of claim 31, further comprising the step of disconnecting the communization link with the

target device so that the target device can be operated independently in a standalone mode.

33. The method of claim 31, wherein the receiving, sending, and preventing steps for a first block of data are all performed prior to the receiving, sending, and preventing steps for a second block of data.

34. The method of claim 31, wherein the data received in the receiving step comprises the authorization in the waiting step.

35. The method of claim 31, wherein the notification providing step comprises the step of notifying a user that the target device may be disconnected.

36. The method of claim 31, wherein, in the providing step, the payment information is provided to the remote server via the Internet, and

wherein, in the receiving step, the data is received from the remote server via the Internet.

37. The method of claim 31, wherein the preventing step comprises the step of erasing the data received in the receiving step or overwriting the data received in the receiving step.

38. The method of claim 31, wherein the receiving, sending, and preventing steps for a first block of data are all performed prior to the receiving, sending, and preventing steps for a second block of data,

wherein the notification providing step comprises the step of notifying a user that the target device may be disconnected,

wherein, in the providing step, the payment information is provided to the remote server via the Internet,

wherein, in the receiving step, the data is received from the remote server via the Internet, and

wherein the preventing step comprises the step of erasing the data received in the receiving step or overwriting the data received in the receiving step.

39. A method of using a computer to write data into a memory for an electronic device, the method comprising the steps of:

loading data into the computer from a data source;

establishing a removable connection between the computer and the memory for the electronic device;

determining if a payment has been made;

writing the data into the memory for the electronic device, via the connection, in response to a determination that a payment has been made;

removing, after completion of the writing step, the removable connection between the computer and the memory for the electronic device; and

operating the electronic device independently of the computer, after completion of the removing step, using the data written in the writing step.

40. The method of claim 39, wherein the memory for the electronic device is housed in a plug-in module.

41. The method of claim 39, wherein the memory for the electronic device is housed in a PCMCIA card.

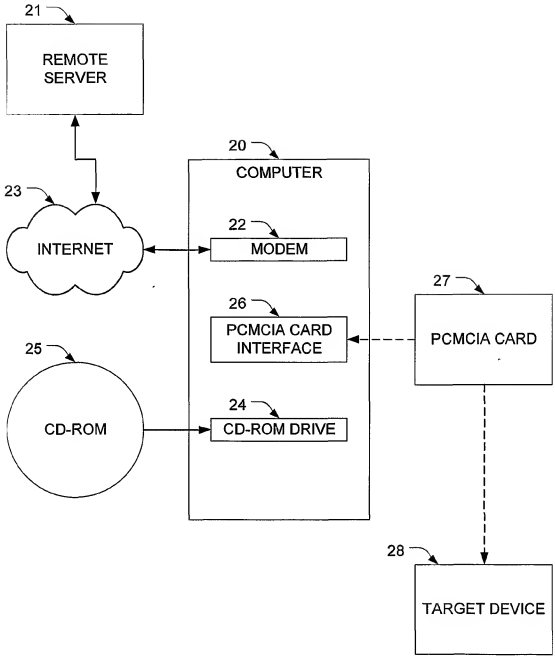

21
REMOTE SERVER
20
COMPUTER
23
INTERNET
22
MODEM
27
26
PCMCIA CARD INTERFACE
PCMCIA CARD
25
CD-ROM
24
CD-ROM DRIVE
28
TARGET DEVICE

FIG. 1

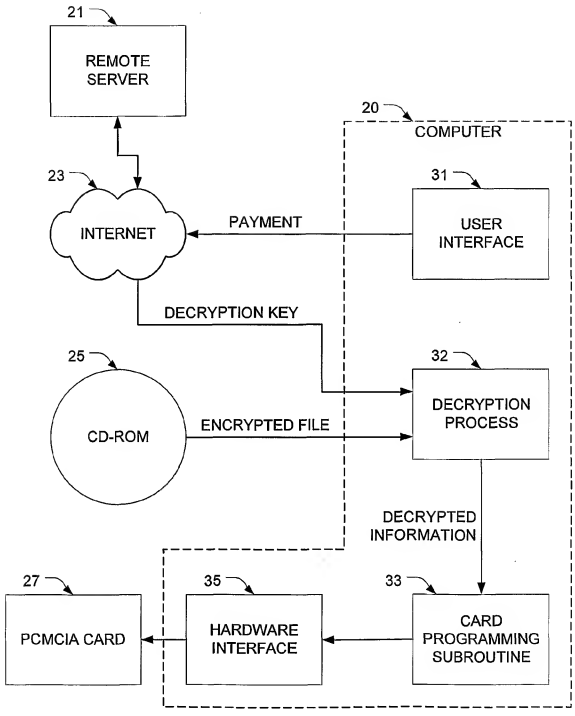

21
REMOTE SERVER
20
COMPUTER
23
INTERNET
PAYMENT
31
USER INTERFACE
DECRYPTION KEY
25
CD-ROM
ENCRYPTED FILE
32
DECRYPTION PROCESS
DECRYPTED INFORMATION
27
PCMCIA CARD
35
HARDWARE INTERFACE
33
CARD PROGRAMMING SUBROUTINE

FIG. 2

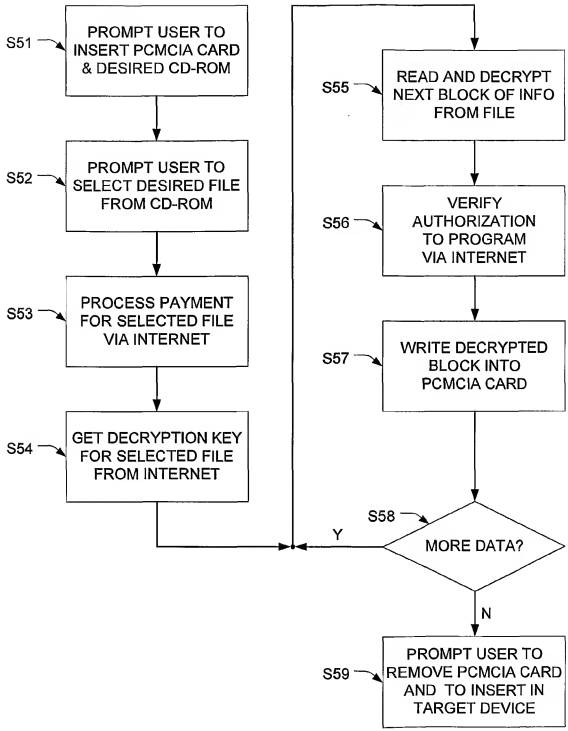

S51
PROMPT USER TO INSERT PCMCIA CARD & DESIRED CD-ROM
S52
PROMPT USER TO SELECT DESIRED FILE FROM CD-ROM
S53
PROCESS PAYMENT FOR SELECTED FILE VIA INTERNET
S54
GET DECRYPTION KEY FOR SELECTED FILE FROM INTERNET
S55
READ AND DECRYPT NEXT BLOCK OF INFO FROM FILE
S56
VERIFY AUTHORIZATION TO PROGRAM VIA INTERNET
S57
WRITE DECRYPTED BLOCK INTO PCMCIA CARD
S58
MORE DATA?
Y
N
S59
PROMPT USER TO REMOVE PCMCIA CARD AND TO INSERT IN TARGET DEVICE

FIG. 3

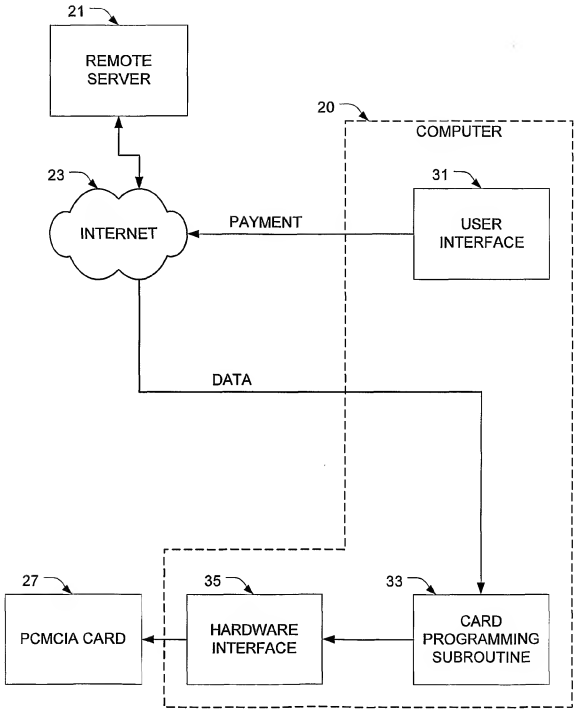

21
REMOTE SERVER
20
COMPUTER
23
INTERNET
PAYMENT
31
USER INTERFACE
DATA
27
PCMCIA CARD
35
HARDWARE INTERFACE
33
CARD PROGRAMMING SUBROUTINE

FIG. 4

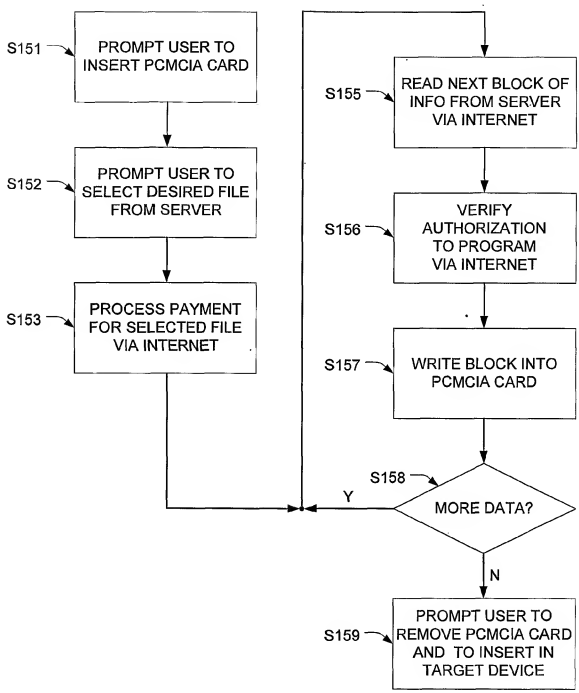

S151
PROMPT USER TO INSERT PCMCIA CARD
S152
PROMPT USER TO SELECT DESIRED FILE FROM SERVER
S153
PROCESS PAYMENT FOR SELECTED FILE VIA INTERNET
S155
READ NEXT BLOCK OF INFO FROM SERVER VIA INTERNET
S156
VERIFY AUTHORIZATION TO PROGRAM VIA INTERNET
S157
WRITE BLOCK INTO PCMCIA CARD
S158
MORE DATA?
Y
N
S159
PROMPT USER TO REMOVE PCMCIA CARD AND TO INSERT IN TARGET DEVICE

FIG. 5

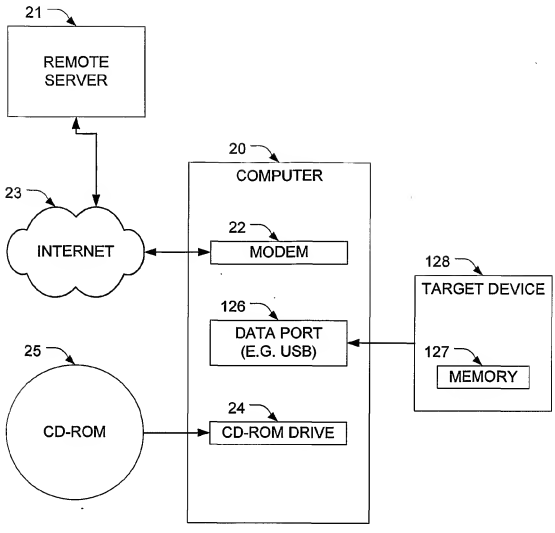

21
REMOTE SERVER
20
COMPUTER
23
INTERNET
22
MODEM
128
TARGET DEVICE
126
DATA PORT (E.G. USB)
127
MEMORY
25
CD-ROM
24
CD-ROM DRIVE

FIG. 6

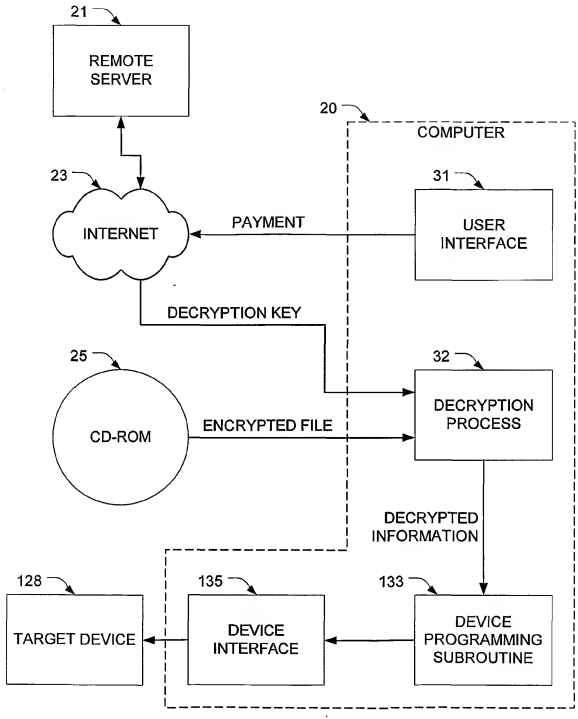

21
REMOTE SERVER
20
COMPUTER
23
INTERNET
PAYMENT
31
USER INTERFACE
DECRYPTION KEY
25
CD-ROM
ENCRYPTED FILE
32
DECRYPTION PROCESS
DECRYPTED INFORMATION
128
TARGET DEVICE
135
DEVICE INTERFACE
133
DEVICE PROGRAMMING SUBROUTINE

FIG. 7

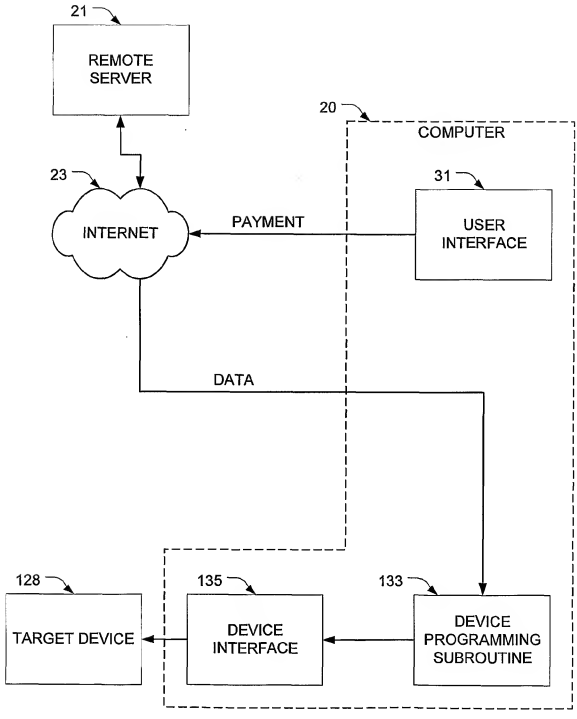

21
REMOTE SERVER
20
COMPUTER
23
INTERNET
PAYMENT
31
USER INTERFACE
DATA
128
TARGET DEVICE
135
DEVICE INTERFACE
133
DEVICE PROGRAMMING SUBROUTINE

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/15672

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 17/60
US CL : 705/51
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 705/51

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,428,685 A (KADOOKA et al) 27 June 1995 (27.06.1995), column 1, lines 11-20 and column 5, lines 39-48 and column 6, lines 21-40 and abstract.	1-41
A	US 5,319,705 A (HALTER et al) 07 June 1994 (07.06.1994), see abstract and column 5, line 28 through column 6, line 45.	1-41
A	US 6,266,416 B1 (SIGBJORNSEN et al) 24 July 2001 (24.07.2001), see abstract.	1-41
A	US 5,761,306 A (LEWIS) 02 June 1998 (02.06.1998), see abstract and column 3, line 15 through column 4, line 42.	1-41

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
01 August 2001 (01.08.2001)	21 SEP 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer Tod Swann
Facsimile No. (703)305-3230	Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)